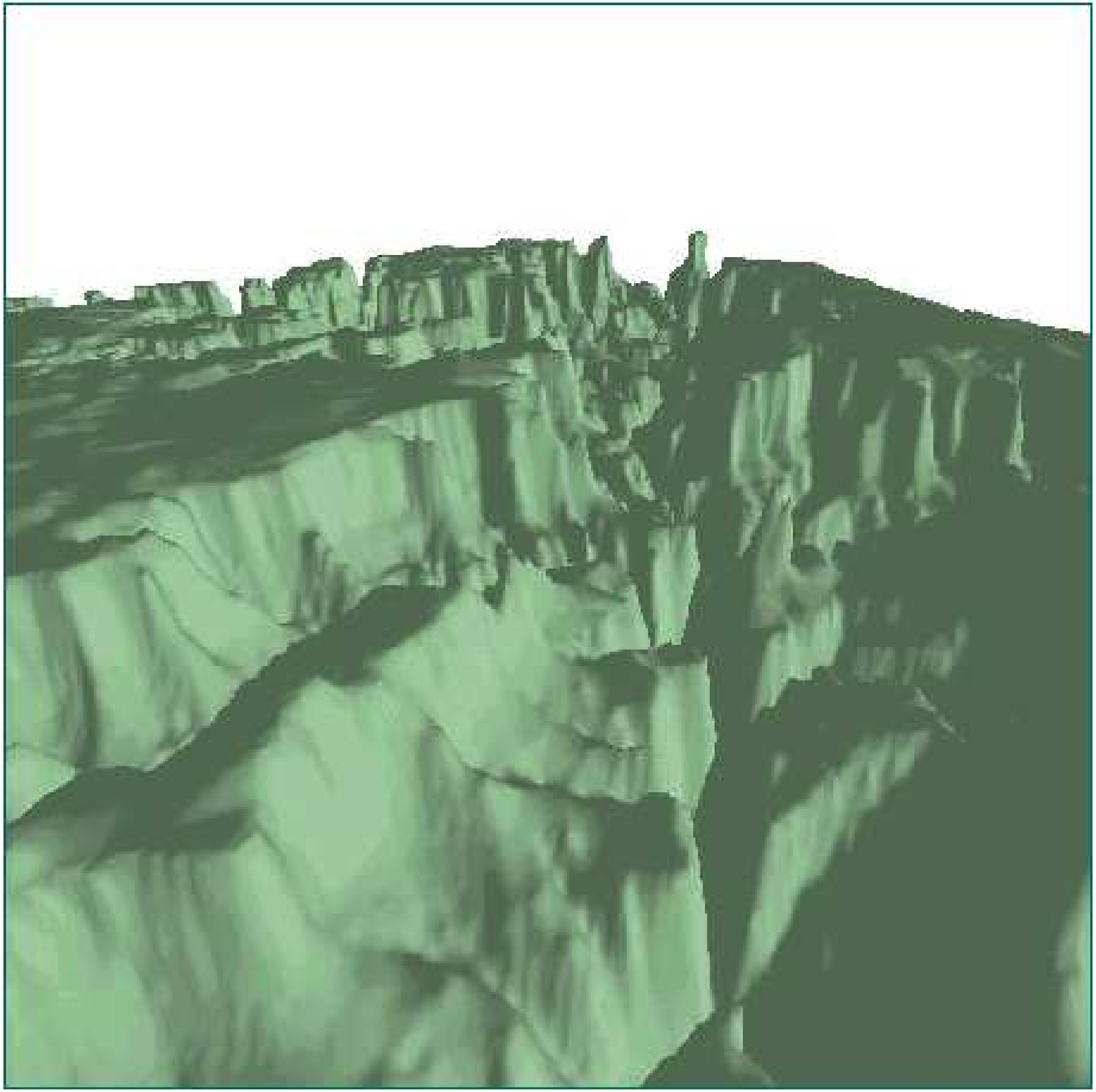
Compressing Large Data Sets with Geometry

PIs: Mathieu Desbrun (USC West)
Ronald DeVore (USC East)
Peter Schröder (Cal Tech)

Large Data Sets with Geometry

- Digital Elevation Maps
- Medical Imagery
- Computer Aided Design
- Reverse Engineering
- Steering Large Scale Computation

Surface: Graph of $2d$- function



Encoders/Decoders

- Server(Encoder) $\longrightarrow$ Client(Decoder)

Encoders/Decoders

- Server(Encoder) $\longrightarrow$ Client(Decoder)
- Encoder: $E : S \rightarrow B(S) = \{b_1, b_2, \ldots, b_n, \ldots\}$

Encoders/Decoders

- Server(Encoder) $\longrightarrow$ Client(Decoder)
- Encoder: $E : S \rightarrow B(S) = \{b_1, b_2, \ldots, b_n, \ldots\}$
- Client asks for surface

Encoders/Decoders

- Server(Encoder) $\longrightarrow$ Client(Decoder)
- Encoder: $E : S \to B(S) = \{b_1, b_2, \ldots, b_n, \ldots\}$
- Client asks for surface
- $E_n(S) := \{b_1, \ldots, b_n\}$

Encoders/Decoders

- Server(Encoder) $\longrightarrow$ Client(Decoder)
- Encoder: $E : S \rightarrow B(S) = \{b_1, b_2, \ldots, b_n, \ldots\}$
- Client asks for surface
- $E_n(S) := \{b_1, \ldots, b_n\}$
- Decoder: $D_n : B \rightarrow S_B$

Encoders/Decoders

- Server(Encoder) $\longrightarrow$ Client(Decoder)
- Encoder: $E : S \longrightarrow B(S) = \{b_1, b_2, \ldots, b_n, \ldots\}$
- Client asks for surface
- $E_n(S) := \{b_1, \ldots, b_n\}$
- Decoder: $D_n : B \longrightarrow S_B$
- Compressed Surface: $D_n(E_n(S)) := S_B$

Encoders/Decoders

- Server(Encoder) $\longrightarrow$ Client(Decoder)
- Encoder: $E : S \rightarrow B(S) = \{b_1, b_2, \ldots, b_n, \ldots\}$
- Client asks for surface
- $E_n(S) := \{b_1, \ldots, b_n\}$
- Decoder: $D_n : B \longrightarrow S_B$
- Compressed Surface: $D_n(E_n(S)) := S_B$
- if resolution is unacceptable client asks for more bits

Desirable Properties:

- Universal: Applies to all S

Desirable Properties:

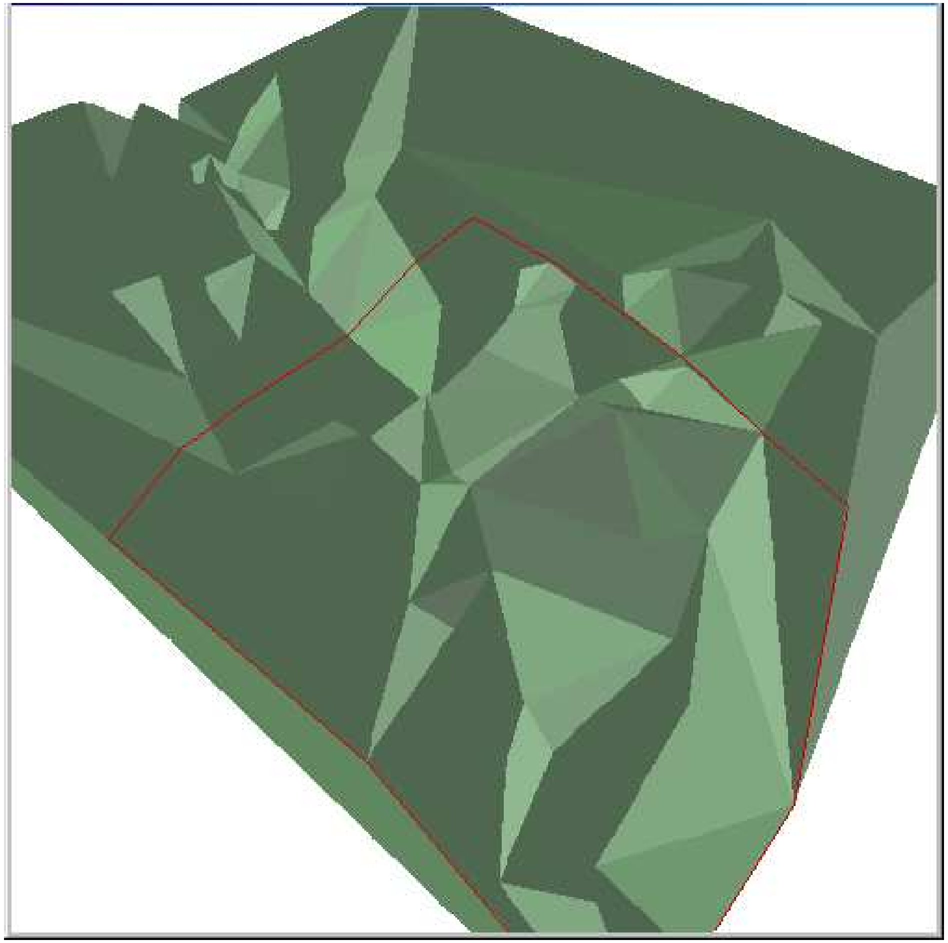
- Universal: Applies to all S
- Progressive: Receiving more bits gives improved resolution

Desirable Properties:

- Universal: Applies to all S
- Progressive: Receiving more bits gives improved resolution
- Burn In: Client selects subregion; new bits only update chosen region

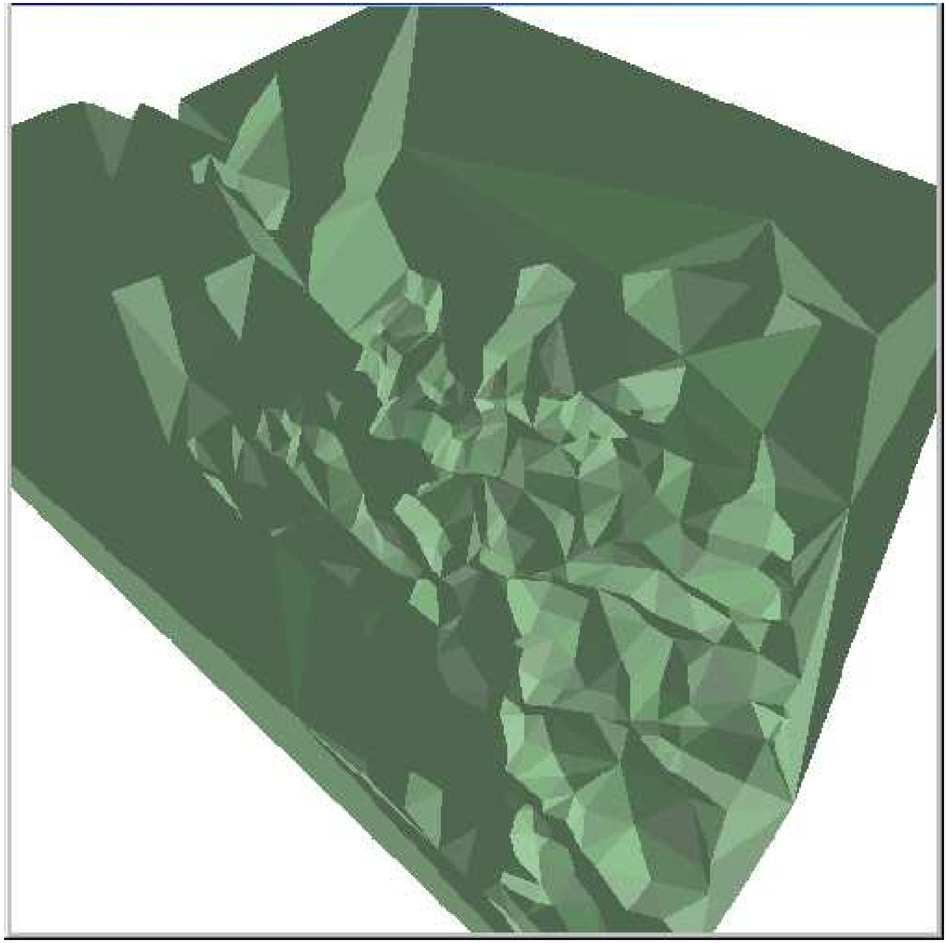
Burn-In

Coarse Approximation



Progressive Burn-In

Selected region for refinement



Desirable Properties:

- Universal: Applies to all S
- Progressive: Receiving more bits gives improved resolution
- Burn In: Client selects subregion; new bits only update chosen region
- Preserve Geometry and Topology

Desirable Properties:

- Universal: Applies to all S
- Progressive: Receiving more bits gives improved resolution
- Burn In: Client selects subregion; new bits only update chosen region
- Preserve Geometry and Topology
- Optimal: performs at best bit rate?
- Image Encoder: Cohen,Dahmen, Daubechies,DeVore
- Burn In: DeVore, Johnson, Sharpley

How can we evaluate algorithms

- Experimental:

  Encoders designed on heuristics

How can we evaluate algorithms

- Experimental:

  Encoders designed on heuristics

- Precise Mathematical Formulation

  Understand rules of game; what it means to be a winner

How can we evaluate algorithms

- Experimental:

  Encoders designed on heuristics

- Precise Mathematical Formulation

  Understand rules of game; what it means to be a winner

- Two essential ingredients

  a. metric ρ to measure distortion

  b. Precise definition of classes K_α to be compressed

Mathematical Formulation

- Distortion: $\rho(S, D_n E_n(S))$

Mathematical Formulation

- Distortion: $\rho(S, D_n E_n(S))$
- Evaluate Performance on a set K of surfaces

$$\delta(K; D_n, E_n) := \sup_{S \in K} \rho(S, D_n E_n(S))$$

Mathematical Formulation

- Distortion: $\rho(S, D_n E_n(S))$
- Evaluate Performance on a set K of surfaces

$$\delta(K; D_n, E_n) := \sup_{S \in K} \rho(S, D_n E_n(S))$$

- Given bit budget n

$$\delta_n(K) := \inf_{E_n, D_n} \delta(K, D_n E_n(S))$$

Mathematical Formulation

- Distortion: $\rho(S, D_n E_n(S))$
- Evaluate Performance on a set K of surfaces

$$\delta(K; D_n, E_n) := \sup_{S \in K} \rho(S, D_n E_n(S))$$

- Given bit budget n

$$\delta_n(K) := \inf_{E_n, D_n} \delta(K, D_n E_n(S))$$

- smallest distortion for the given bit budget

(Near) Optimal Encoding for K

- optimal

$$\delta(K, D_n E_n(S)) = \delta_n(K)$$

(Near) Optimal Encoding for K

- optimal

$$\delta(K, D_n E_n(S)) = \delta_n(K)$$

- near optimal

$$\delta(K, D_n E_n(S)) \leq C\delta_n(K)$$

(Near) Optimal Encoding for K

- optimal

$$\delta(K, D_n E_n(S)) = \delta_n(K)$$

- near optimal

$$\delta(K, D_n E_n(S)) \leq C\delta_n(K)$$

- Typically: $\delta_n(K) \approx n^{-s}$ for some $s > 0$

(Near) Optimal Encoding for K

- optimal

$$\delta(K, D_n E_n(S)) = \delta_n(K)$$

- near optimal

$$\delta(K, D_n E_n(S)) \leq C\delta_n(K)$$

- Typically: $\delta_n(K) \approx n^{-s}$ for some $s > 0$
- Game: Find encoder/decoder E/D: for all values of n and all classes K_α, encoder is near optimal

Optimal Encoding: Kolmogorov Entropy

- Given $\epsilon > 0$

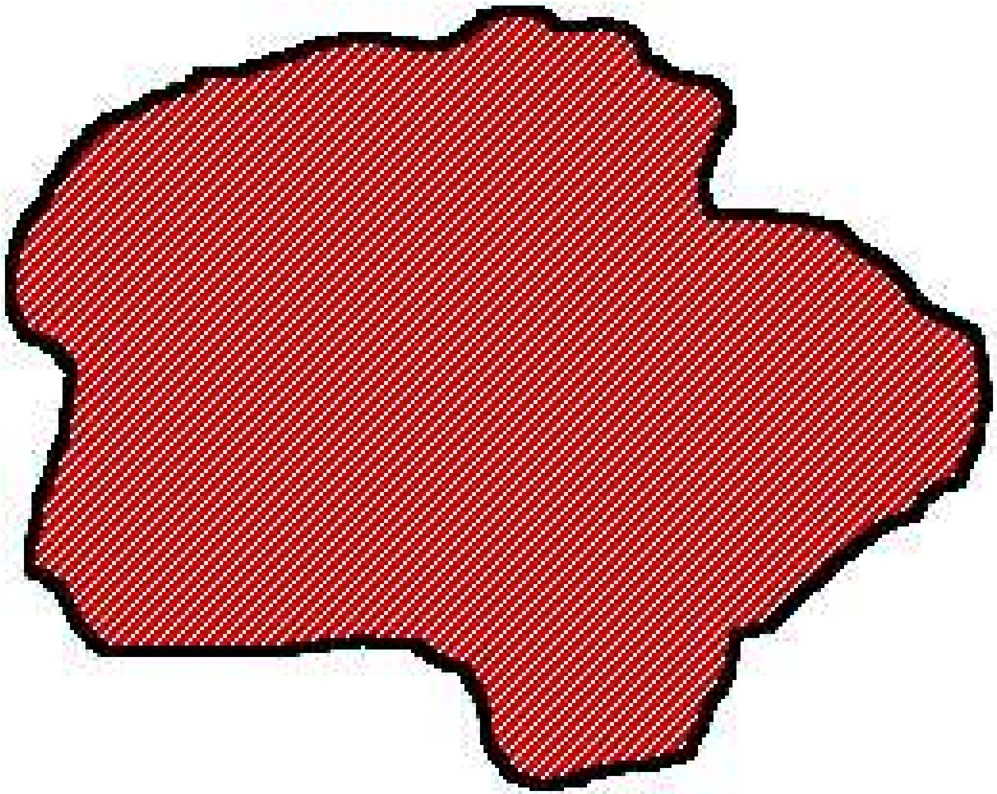
Optimal Encoding: Kolmogorov Entropy

- Given $\epsilon > 0$
- Minimal ϵ cover: $K \subset \cup_{i=1}^{N_\epsilon} \mathcal{B}(S_i, \epsilon)$

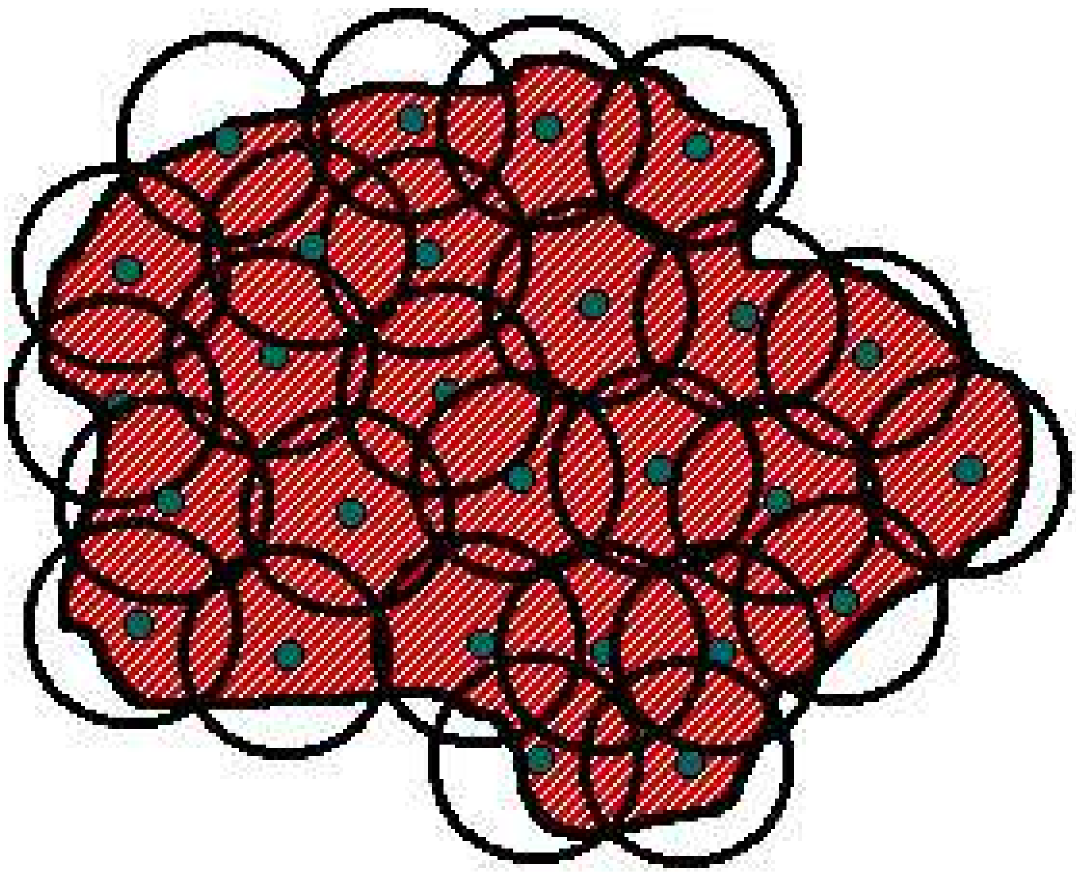
Optimal Encoding: Kolmogorov Entropy

- Given $\epsilon > 0$
- Minimal ϵ cover: $K \subset \cup_{i=1}^{N_\epsilon} \mathcal{B}(S_i, \epsilon)$
- Kolmogorov Entropy $H_\epsilon(K) := \log_2 N_\epsilon(K)$

Covering



Covering



Kolmogorov Entropy

- Given $\epsilon > 0$
- Minimal ϵ cover: $K \subset \cup_{i=1}^{N_\epsilon} \mathcal{B}(S_i, \epsilon)$
- Kolmogorov Entropy $H_\epsilon(K) := \log_2 N_\epsilon(K)$

- $\delta_n(K) = \inf\{\epsilon : \ H_\epsilon(K) \leq n\}$

Optimal Encoding: Kolmogorov Entropy

- Given $\epsilon > 0$
- Minimal ϵ cover: $K \subset \cup_{i=1}^{N_\epsilon} \mathcal{B}(x_i, \epsilon)$
- $\delta_n(K) = \inf\{\epsilon : \; H_\epsilon(K) \leq n$
- Kolmogorov entropy of K gives our benchmark
- Usually not practical encoder

The Issues

1. The metric
2. The classes
3. Determine Entropy of Classes
4. Build near optimal Encoders/Decoders

Possible Metrics

- L_2 = Least squares not appropriate

Possible Metrics

- L_2 = Least squares not appropriate
- L_∞ better

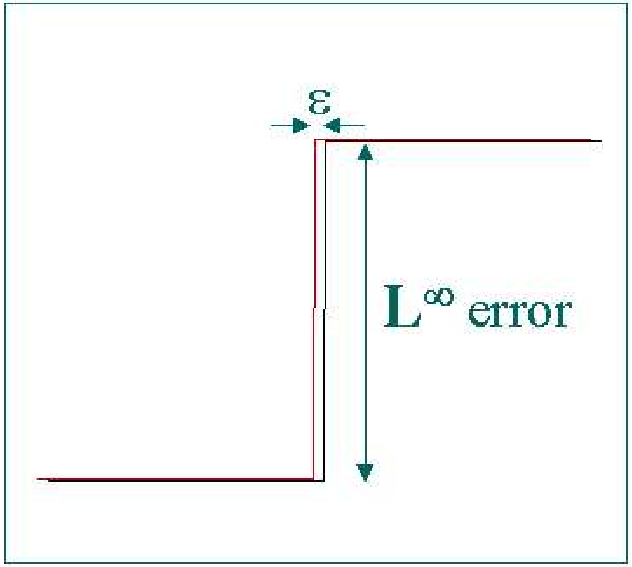
Possible Metrics

- L_2 = Least squares not appropriate
- L_∞ better
- Hausdorff better yet

Comparison of Metric

- Offset by a lateral error of ϵ, L^∞ error may be huge

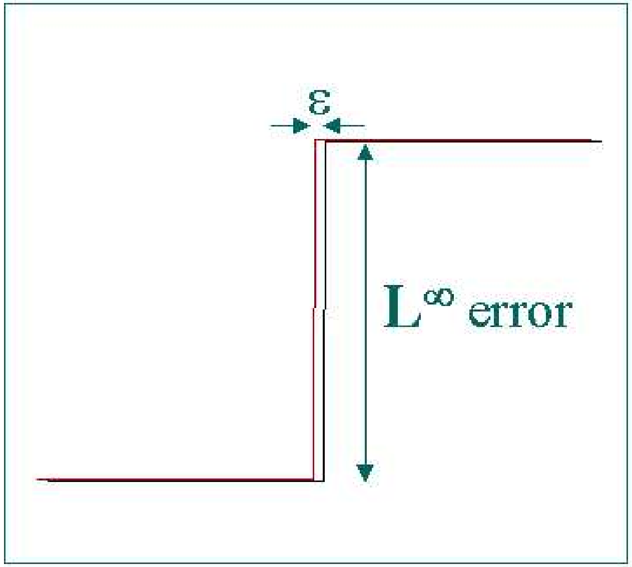
L^∞ metric error



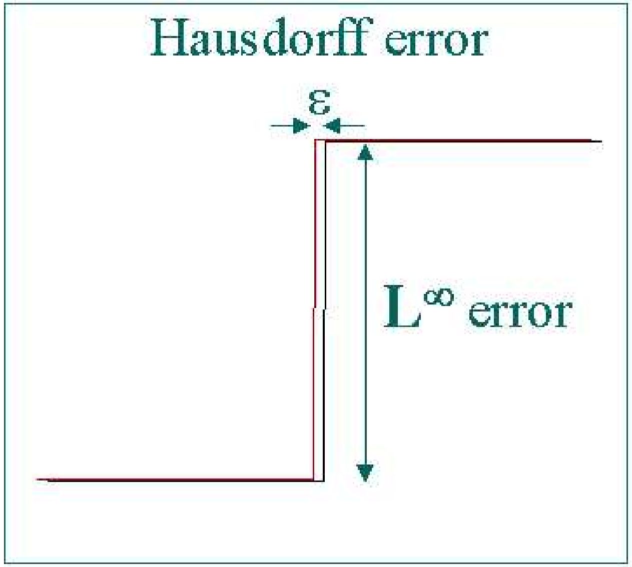
Comparison of Metric

- Offset by a lateral error of ϵ, L^∞ error may be huge
- Hausdorff error is as expected

L^∞ metric error



Hausdorff metric error



Possible Metrics

- L_2 = Least squares not appropriate
- L_∞ better
- Hausdorff better yet
- $d(S, S') := \sup_{x \in S} \operatorname{dist}(x, S')$

Possible Metrics

- L_2 = Least squares not appropriate
- L_∞ better
- Hausdorff better yet
- $d(S, S') := \sup_{x \in S} \operatorname{dist}(x, S')$
- $\delta_H(S, S') := d(S, S') + d(S', S)$

Other Possible Metrics

- metrics to incorporate geometry/topology

Other Possible Metrics

- metrics to incorporate geometry/topology
- metrics to incorporate line of sight

Classification of Surfaces

- Classical Approach is smoothness spaces - too isotropic

Classification of Surfaces

- Classical Approach is smoothness spaces - too isotropic
- Classify according to complexity

Classification of Surfaces

- Classical Approach is smoothness spaces - too isotropic
- Classify according to complexity
- Complexity of Topology

Classification of Surfaces

- Classical Approach is smoothness spaces - too isotropic
- Classify according to complexity
- Complexity of Topology
- Complexity of Geometry

Classification of Surfaces

- Classical Approach is smoothness spaces - too isotropic
- Classify according to complexity
- Complexity of Topology
- Complexity of Geometry
- Differential Geometry to play crucial role

Classify Surfaces

- Critical points

Classify Surfaces

- Critical points
- ridge curves

Classify Surfaces

- Critical points
- ridge curves
- drainage curves

Classify Surfaces

- Critical points
- ridge curves
- drainage curves
- level curves

Classify Surfaces

- Critical points
- ridge curves
- drainage curves
- level curves
- plateaus

Classify Surfaces

- Critical points
- ridge curves
- drainage curves
- level curves
- plateaus
- smooth regions

Classify on basis of the complexity of these curves

- each curve γ has a smoothness parameter $\alpha(\gamma)$: measures smoothness of γ in certain Besov spaces B^{α} corresponding to L_{∞} (or Hausdorff)

Classify on basis of the complexity of these curves

- each curve γ has a smoothness parameter $\alpha(\gamma)$: measures smoothness of γ in certain Besov spaces B^α corresponding to L_∞ (or Hausdorff)
- We know how to do this by results in univariate approximation - free knot splines; n-term wavelets; optimal encoding

Classify on basis of the complex-ity of these curves

- each curve γ has a smoothness parameter $\alpha(\gamma)$: measures smoothness of γ in certain Besov spaces B^α corresponding to L_∞ (or Hausdorff)
- We know how to do this by results in univariate approximation - free knot splines; n-term wavelets; optimal encoding
- $\delta_n(U(B^\alpha)) \leq Cn^{-\alpha}$

Classify on basis of the complexity of these curves

- each curve γ has a smoothness parameter $\alpha(\gamma)$: measures smoothness of γ in certain Besov spaces B^α corresponding to L_∞ (or Hausdorff)
- We know how to do this by results in univariate approximation - free knot splines; n-term wavelets; optimal encoding
- $\delta_n(U(B^\alpha)) \leq Cn^{-\alpha}$
- Smoothness β between curves: how effectively can we approximate by interpolation

Classify on basis of the complexity of these curves

- each curve γ has a smoothness parameter $\alpha(\gamma)$: measures smoothness of γ in certain Besov spaces B^α corresponding to L_∞ (or Hausdorff)
- We know how to do this by results in univariate approximation - free knot splines; n-term wavelets; optimal encoding
- $\delta_n(U(B^\alpha)) \leq Cn^{-\alpha}$
- Smoothness β between curves: how effectively can we approximate by interpolation
- Classify S on basis of behavior of α and β

Classify on basis of the complexity of these curves

- each curve γ has a smoothness parameter $\alpha(\gamma)$: measures smoothness of γ in certain Besov spaces B^α corresponding to L_∞ (or Hausdorff)
- We know how to do this by results in univariate approximation - free knot splines; n-term wavelets; optimal encoding
- $\delta_n(U(B^\alpha)) \leq Cn^{-\alpha}$
- Smoothness β between curves: how effectively can we approximate by interpolation
- Classify S on basis of behavior of α and β
- Not ready to formulate this

Determine Entropy of Classes $K_{\alpha,\beta}$ in Hausdorff Metric

- Albert Cohen, Wolfgang Dahmen, Ingrid Daubechies

Determine Entropy of Classes $K_{\alpha,\beta}$ in Hausdorff Metric

- Albert Cohen, Wolfgang Dahmen, Ingrid Daubechies
- $\delta_N(K) \leq Cn^{-1}$ for K ball in $C(\Omega)$, $\Omega \subset \mathbb{R}^d$, $N = n^{d-1} \log n$

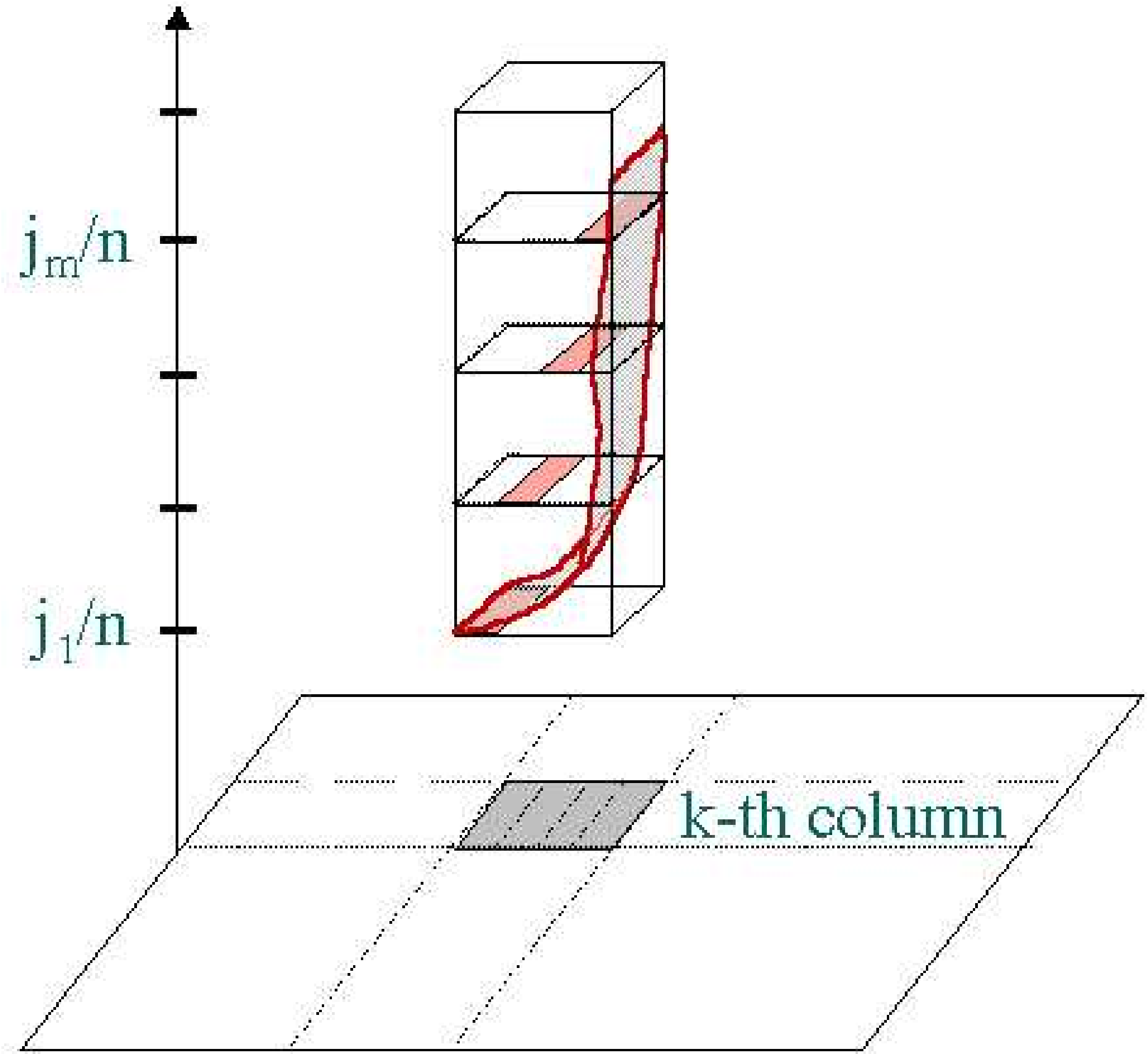
Determine Entropy of Classes $K_{\alpha,\beta}$ in Hausdorff Metric

- Albert Cohen, Wolfgang Dahmen, Ingrid Daubechies
- $\delta_N(K) \leq Cn^{-1}$ for K ball in $C(\Omega)$, $\Omega \subset \mathbb{R}^d$, $N = n^{d-1} \log n$
- $\delta_n(K) \leq Cn^{-1}$ for K ball in $\mathrm{BV}(\Omega)$, $\Omega \subset \mathbb{R}$

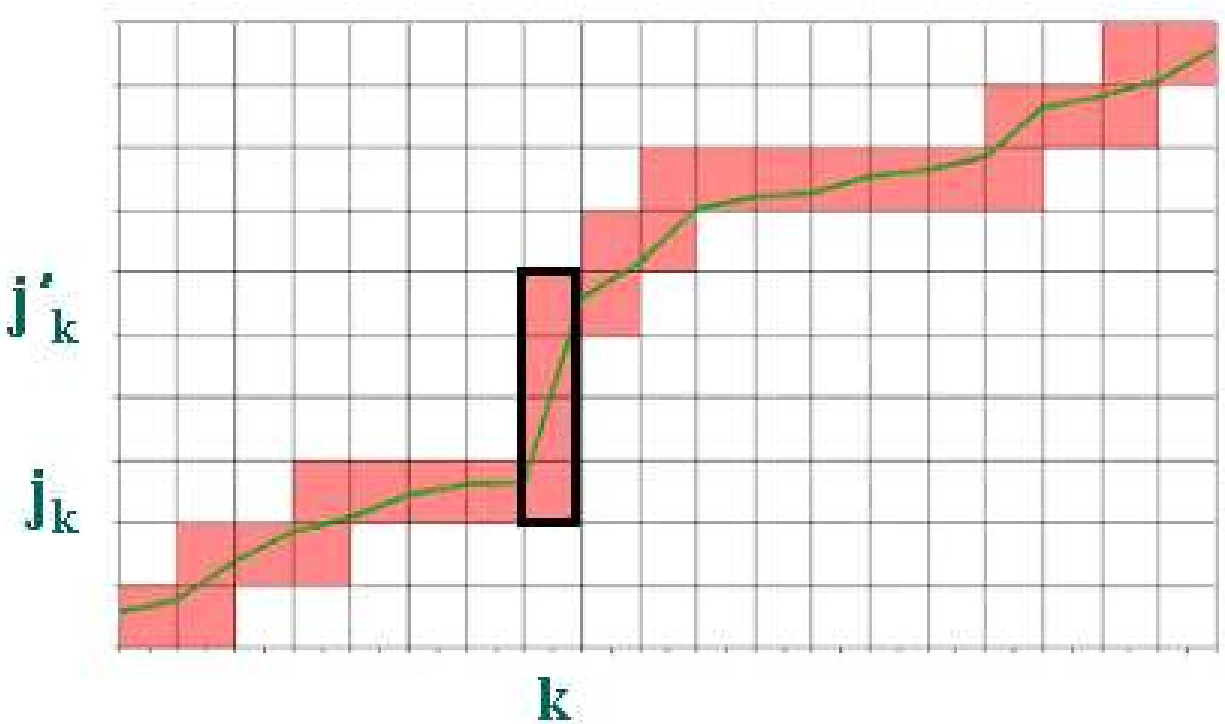
Determine Entropy of Classes $K_{\alpha,\beta}$ in Hausdorff Metric

- Albert Cohen, Wolfgang Dahmen, Ingrid Daubechies
- $\delta_N(K) \leq Cn^{-1}$ for K ball in $C(\Omega)$, $\Omega \subset I\!R^d$, $N = n^{d-1} \log n$
- $\delta_n(K) \leq Cn^{-1}$ for K ball in $\mathrm{BV}(\Omega)$, $\Omega \subset I\!R$
- $\delta_n(K) \leq Cn^{-2}$ for K class of continuous convex in $d = 1$.

box dimension



box dimension



Designing Near Optimal Progressive Encoders

- Extract critical points and curves capturing geometry and topology (wire mesh)

Designing Near Optimal Progressive Encoders

- Extract critical points and curves capturing geometry and topology (wire mesh)
- Prioritize these curves

Designing Near Optimal Progressive Encoders

- Extract critical points and curves capturing geometry and topology (wire mesh)
- Prioritize these curves
- Select a few highest priority curves

Designing Near Optimal Progressive Encoders

- Extract critical points and curves capturing geometry and topology (wire mesh)
- Prioritize these curves
- Select a few highest priority curves
- first bits for these selected curves (first wire mesh)

Designing Near Optimal Progressive Encoders

- Extract critical points and curves capturing geometry and topology (wire mesh)
- Prioritize these curves
- Select a few highest priority curves
- first bits for these selected curves (first wire mesh)
- choose additional next priority curves

Designing Near Optimal Progressive Encoders

- Extract critical points and curves capturing geometry and topology (wire mesh)
- Prioritize these curves
- Select a few highest priority curves
- first bits for these selected curves (first wire mesh)
- choose additional next priority curves
- predict these curves from first wire mesh: generate smooth surface to first wire mesh

Designing Near Optimal Progressive Encoders

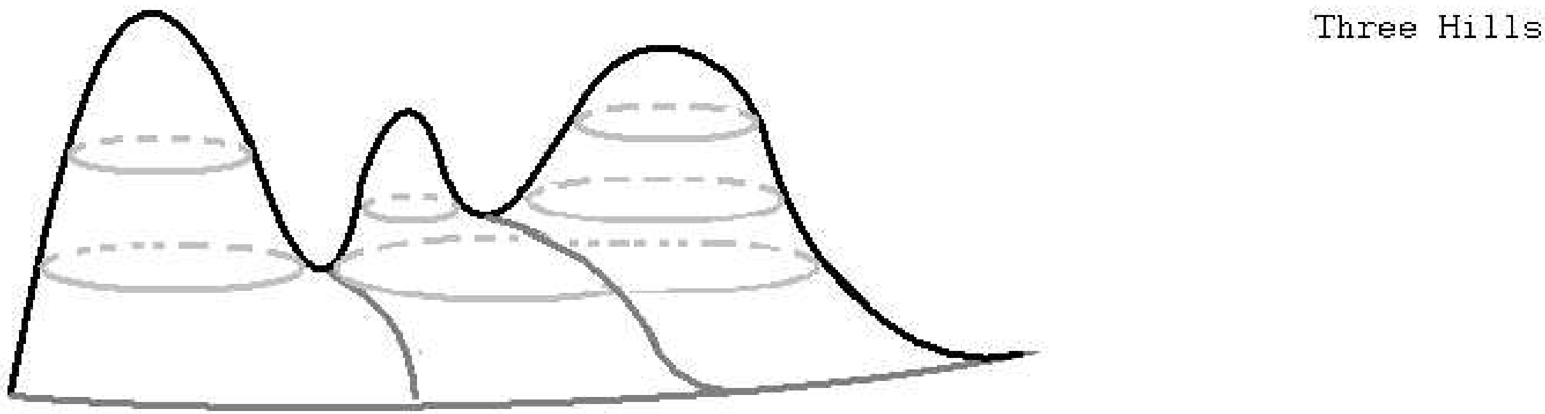
- Extract critical points and curves capturing geometry and topology (wire mesh)
- Prioritize these curves
- Select a few highest priority curves
- first bits for these selected curves (first wire mesh)
- choose additional next priority curves
- predict these curves from first wire mesh: generate smooth surface to first wire mesh
- update bits for all selected curves (second wire mesh)

Designing Near Optimal Progressive Encoders

- Extract critical points and curves capturing geometry and topology (wire mesh)
- Prioritize these curves
- Select a few highest priority curves
- first bits for these selected curves (first wire mesh)
- choose additional next priority curves
- predict these curves from first wire mesh: generate smooth surface to first wire mesh
- update bits for all selected curves (second wire mesh)
- continue

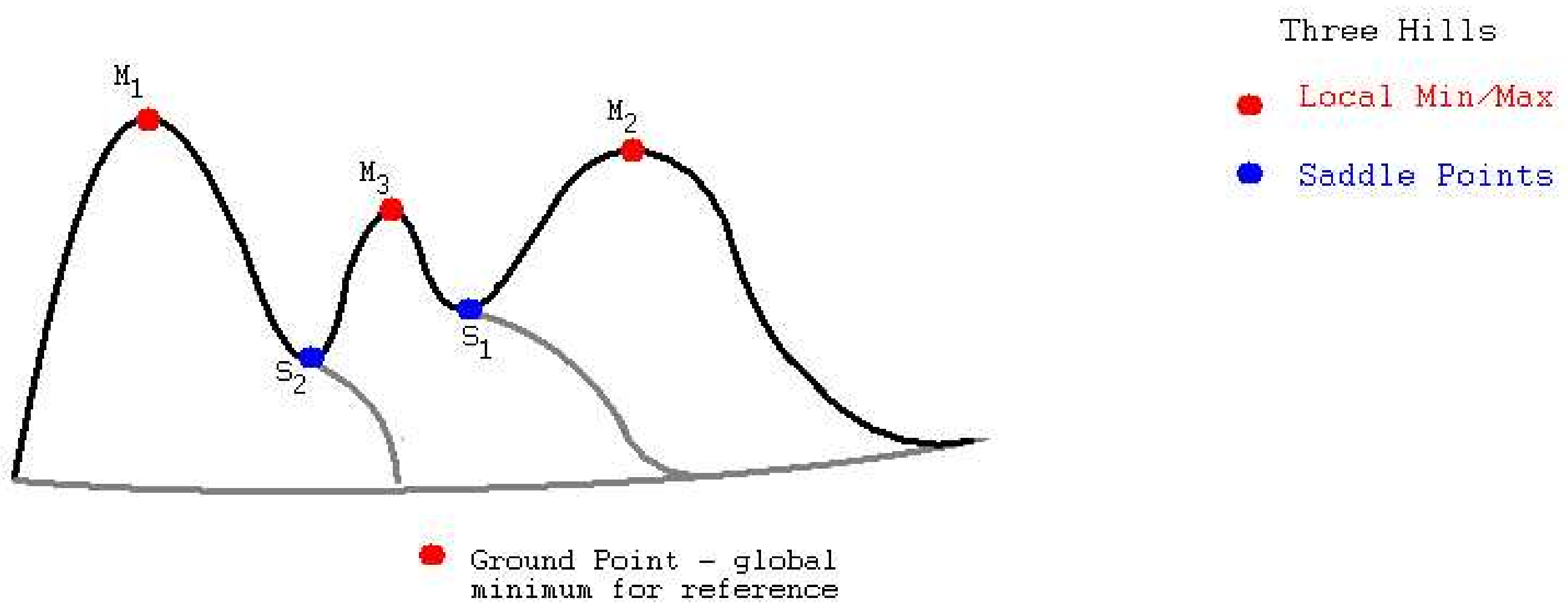
Morse structure and Reeb graphs

Original terrain



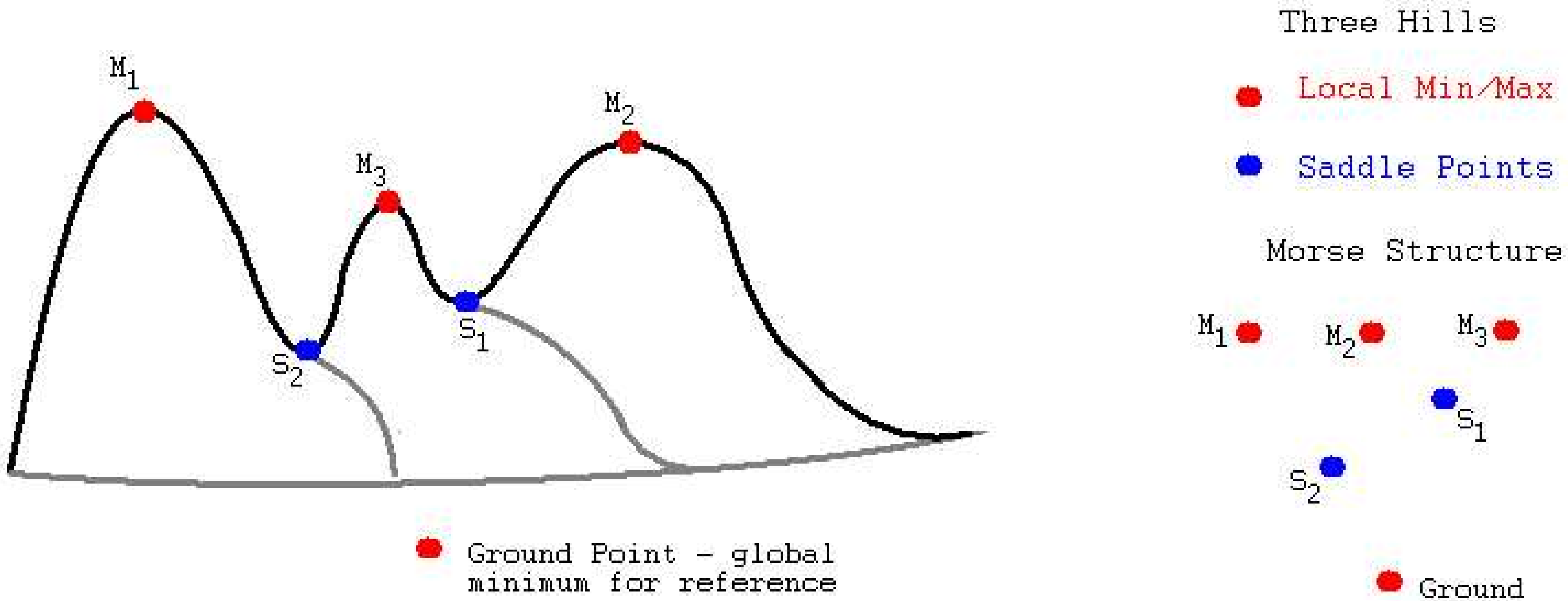
Morse structure and Reeb graphs

Select critical points



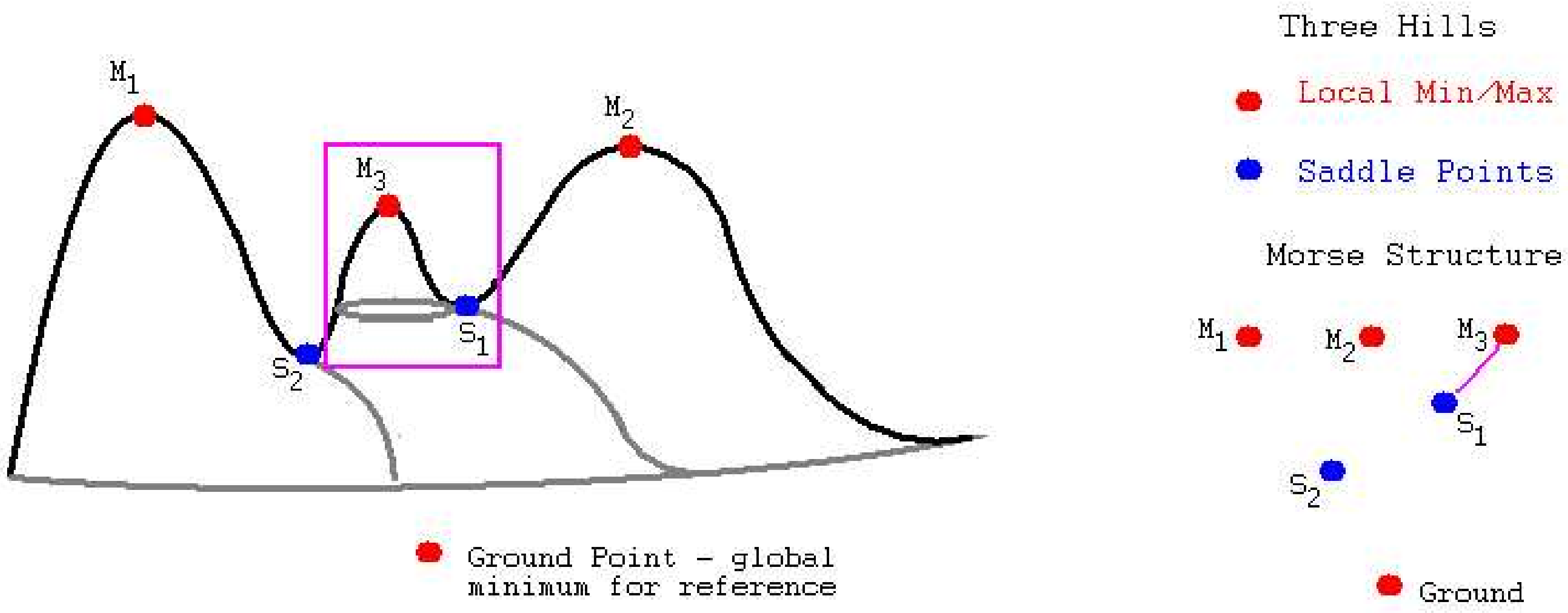
Morse structure and Reeb graphs

Represent as graph



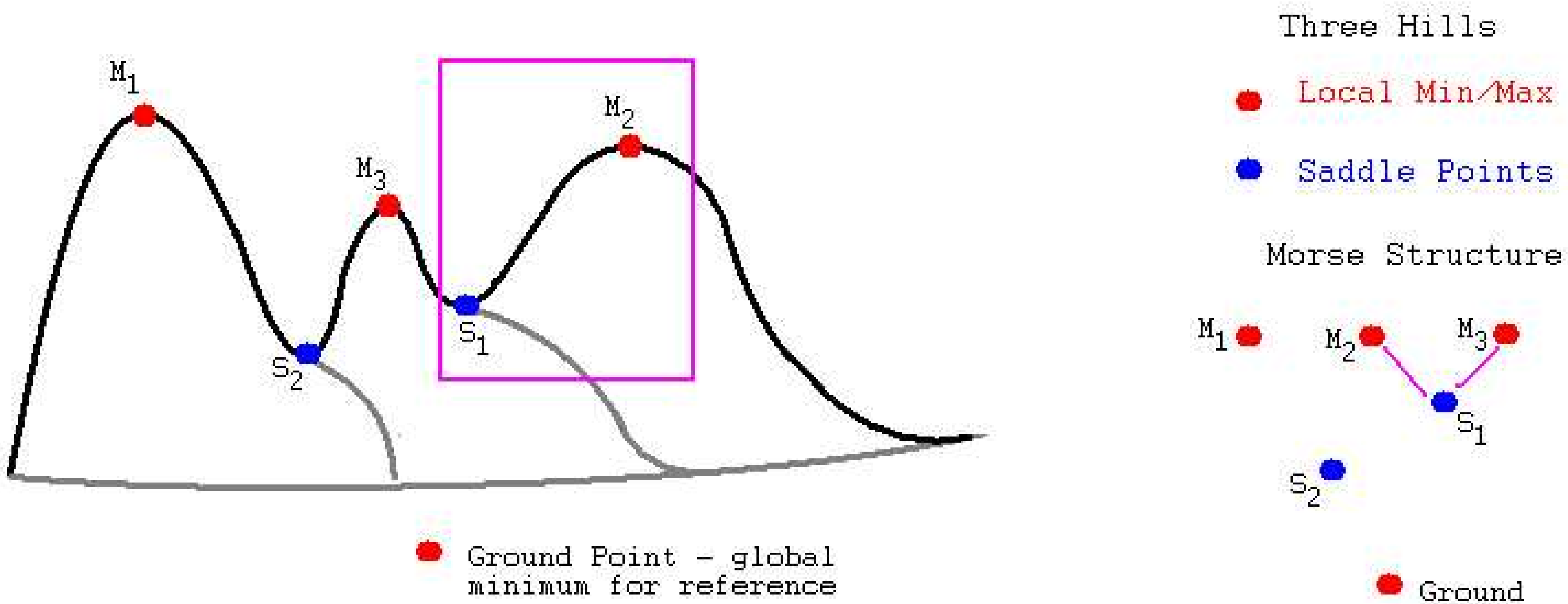
Morse structure and Reeb graphs

Edge represents monotone region



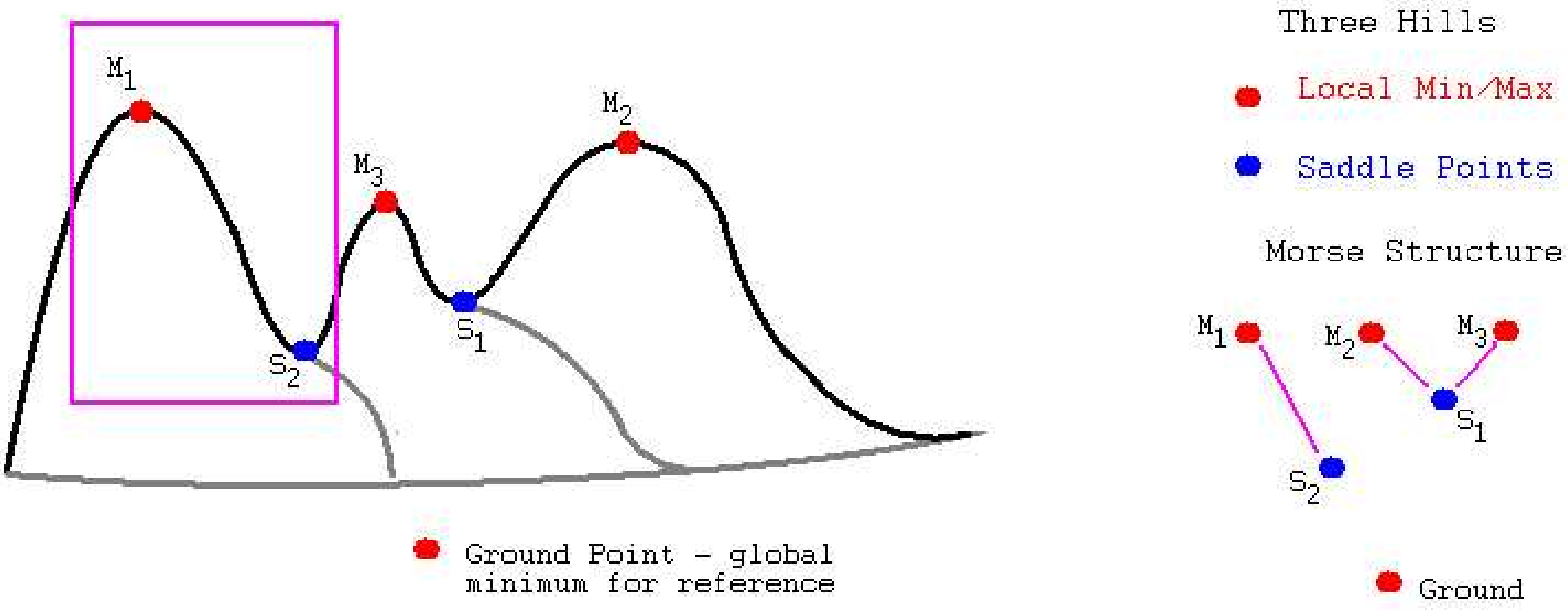
Morse structure and Reeb graphs

Edge represents monotone region



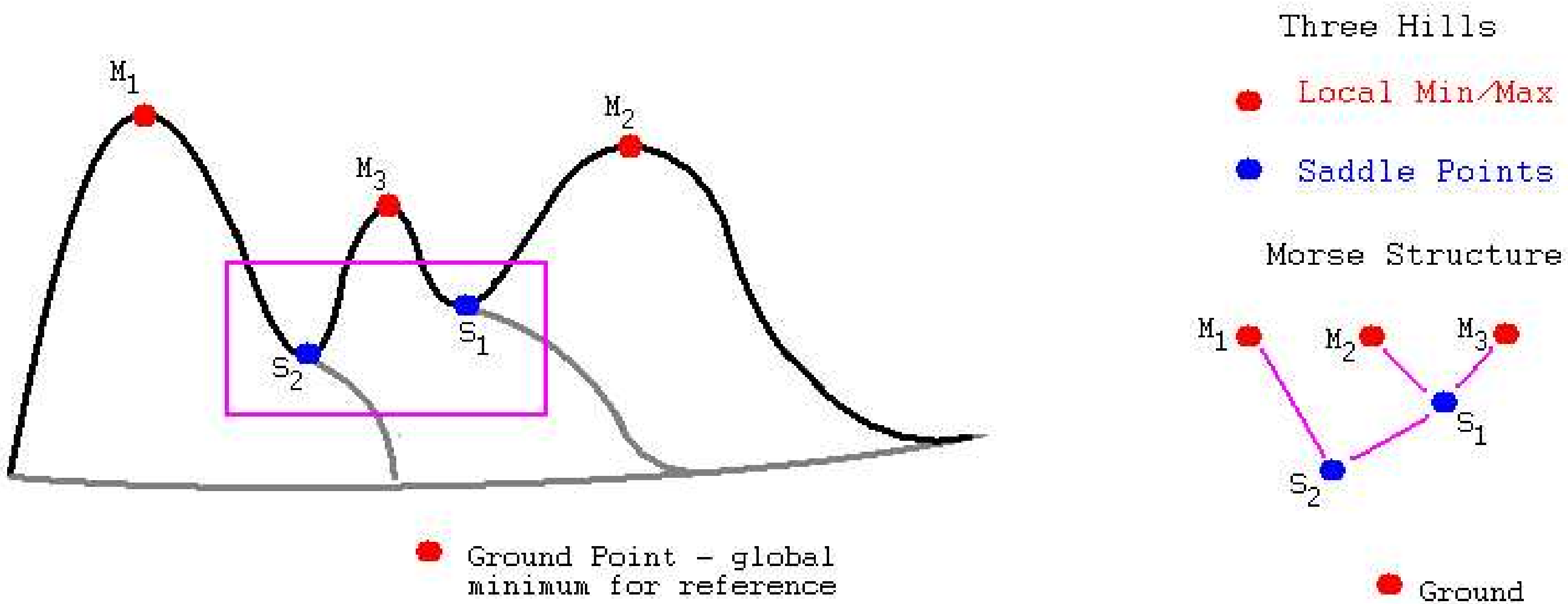
Morse structure and Reeb graphs

Edge represents monotone region



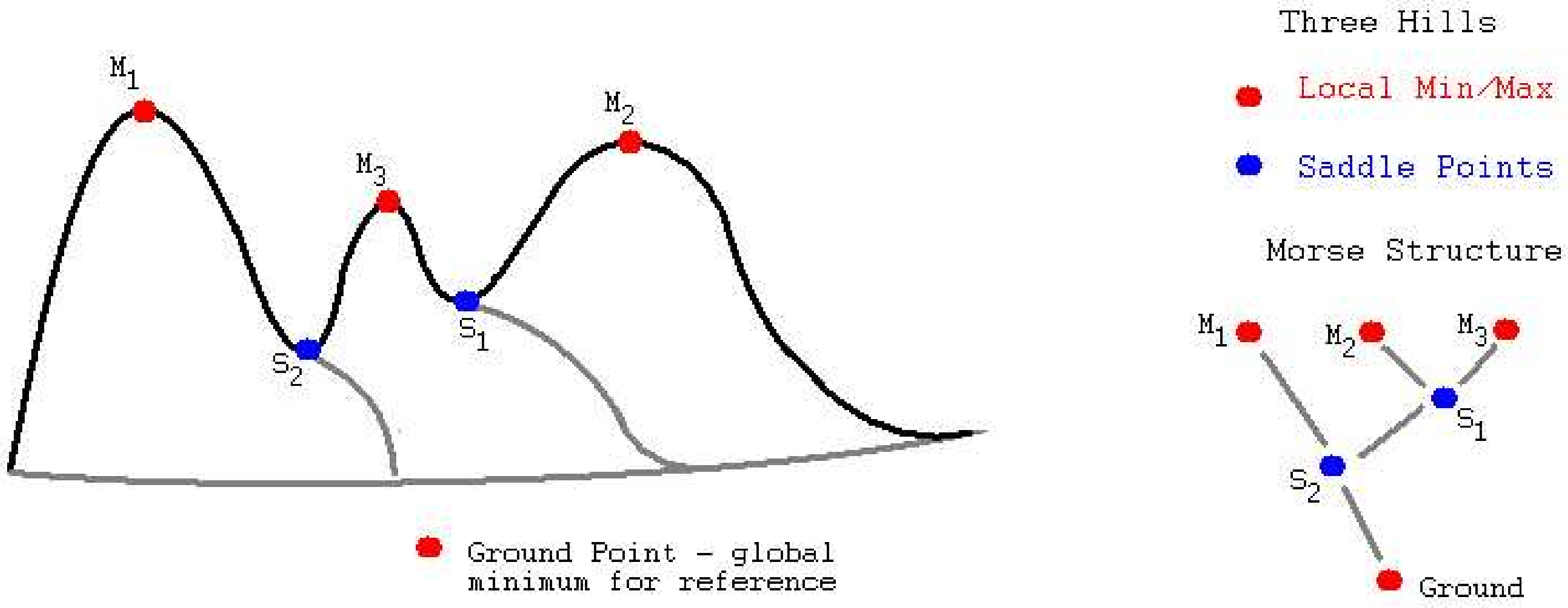
Morse structure and Reeb graphs

monotone region is a washer



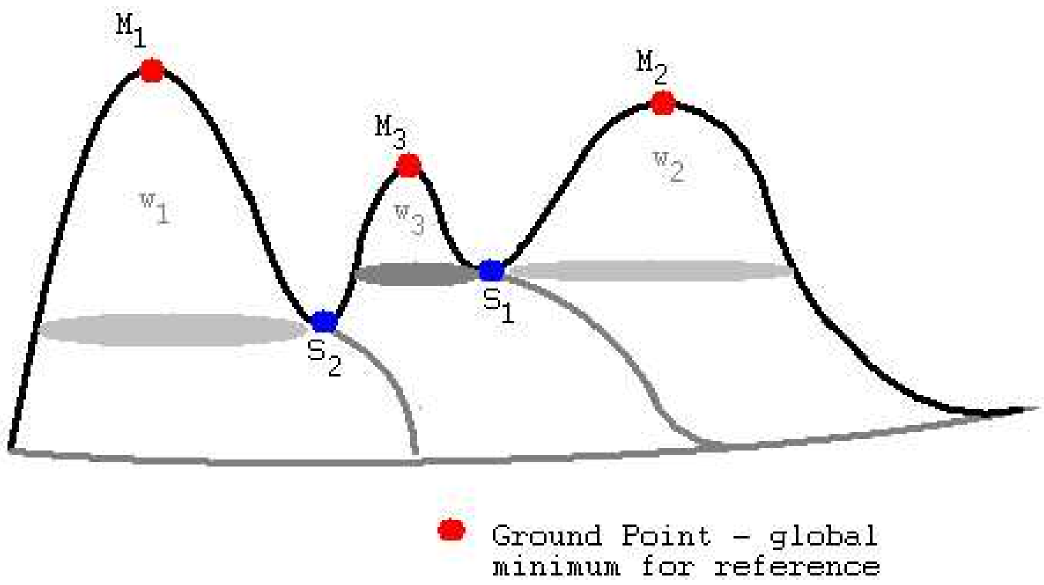
Morse structure and Reeb graphs

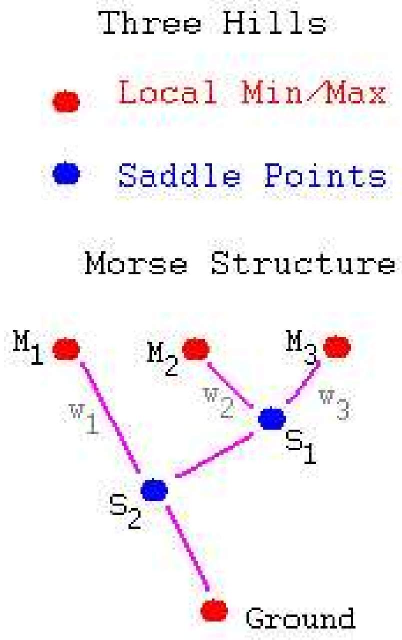
ground reference



Morse structure and Reeb graphs

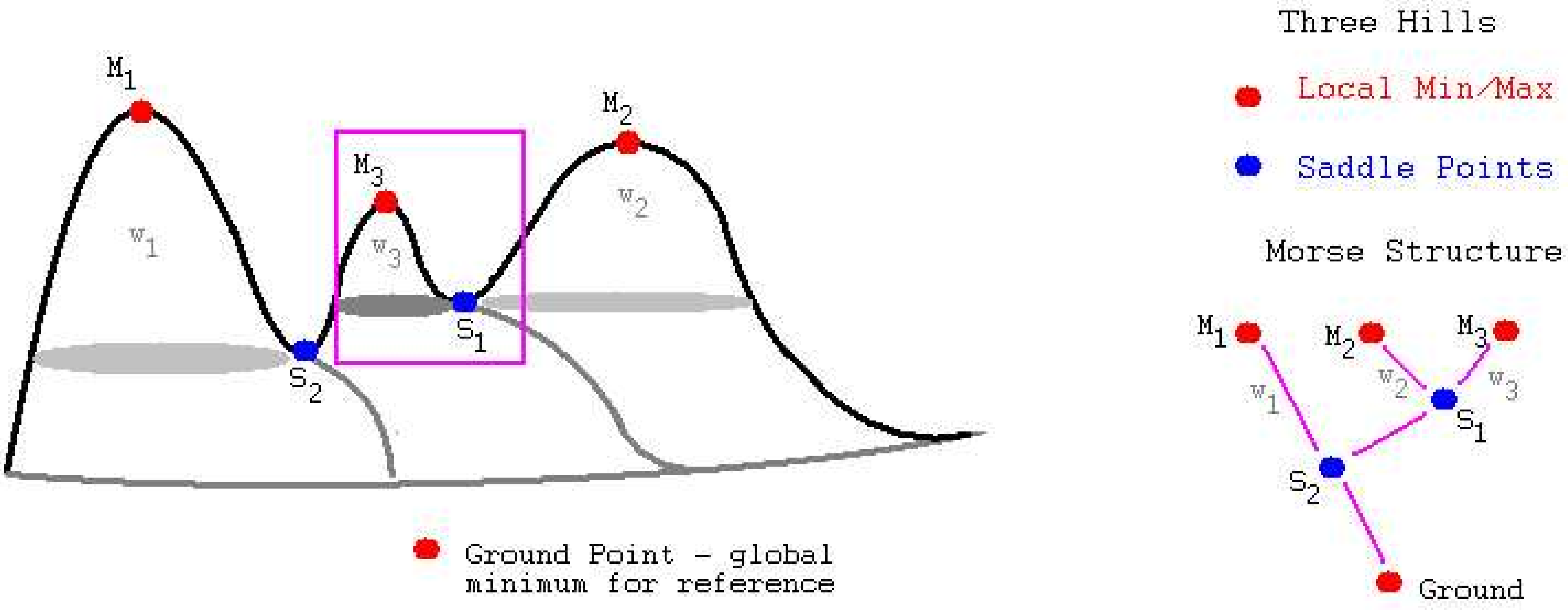
Each saddle point gives level curve





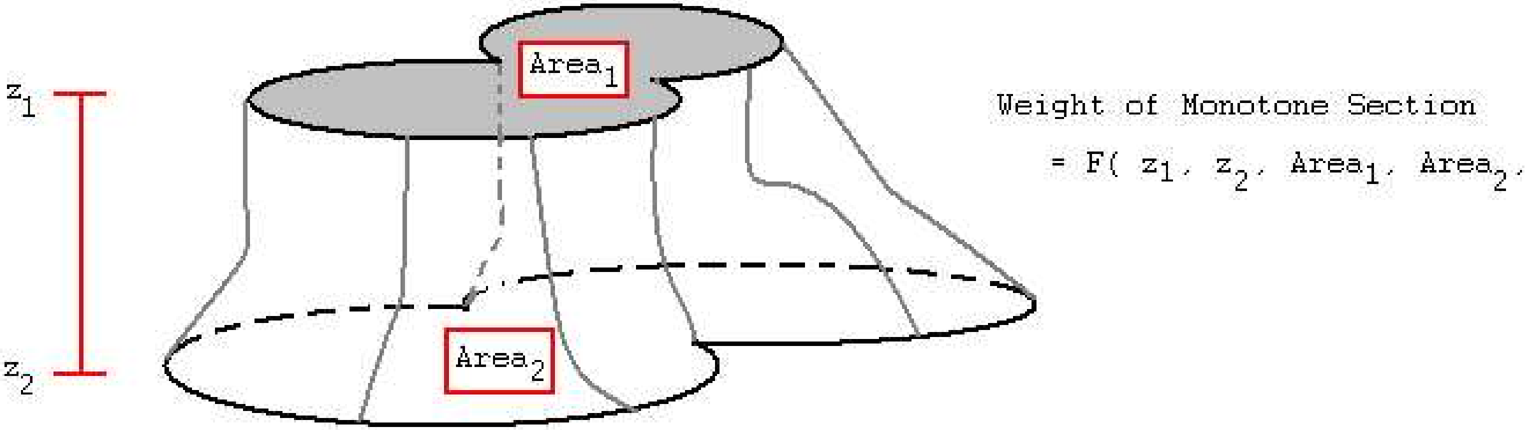
Morse structure and Reeb graphs

prioritize by assigning weights



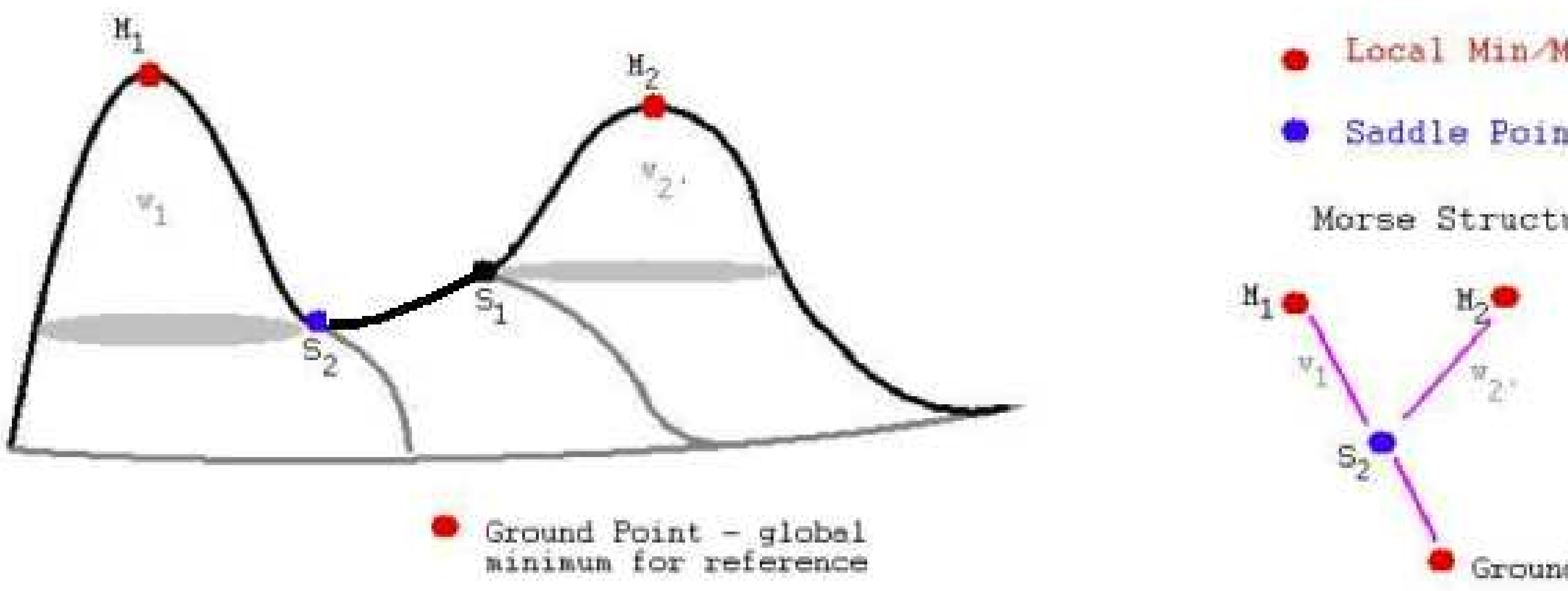
Morse structure and Reeb graphs

Definition of weight



Morse structure and Reeb graphs

removing low priority sections



Add Geometry

- Morse structure captures topology

Add Geometry

- Morse structure captures topology
- Use curvature (shape operator) to select and priortize geometry

Add Geometry

- Morse structure captures topology
- Use curvature (shape operator) to select and priortize geometry
- embed geometry into graph

Encoding Curves

- multiscale methods (CAGD subdivision) using piecewise polynomials

Encoding Curves

- multiscale methods (CAGD subdivision) using piecewise polynomials
- interpolatory

Encoding Curves

- multiscale methods (CAGD subdivision) using piecewise polynomials
- interpolatory
- shape preserving: monotonicity, convexity, etc

Encoding Curves

- multiscale methods (CAGD subdivision) using piecewise polynomials
- interpolatory
- shape preserving: monotonicity, convexity, etc
- optimal encoder for many classes (Besov balls):

$$\delta_n(K_\alpha) \approx n^{-\alpha}, \quad K_\alpha := U(B^\alpha(L_\tau))$$

Encoding Curves

- multiscale methods (CAGD subdivision) using piecewise polynomials
- interpolatory
- shape preserving: monotonicity, convexity, etc
- optimal encoder for many classes (Besov balls):

$$\delta_n(K_\alpha) \approx n^{-\alpha}, \quad K_\alpha := U(B^\alpha(L_\tau))$$

- K_1 includes all curves with finite arc length

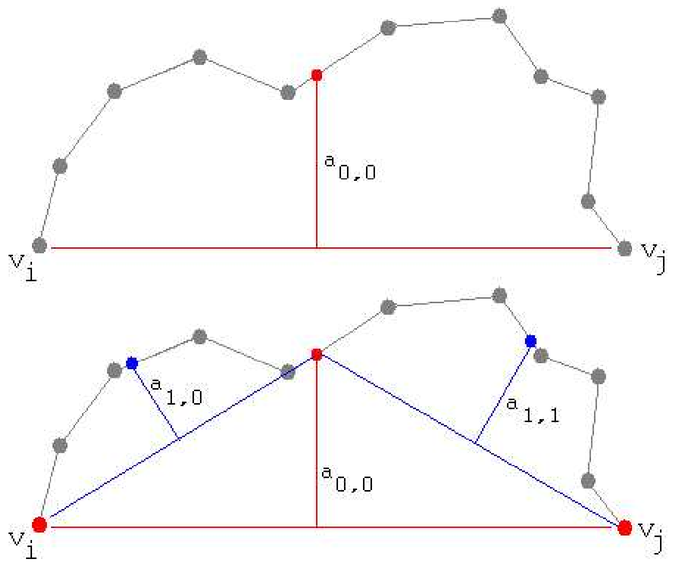
Encoding Curves

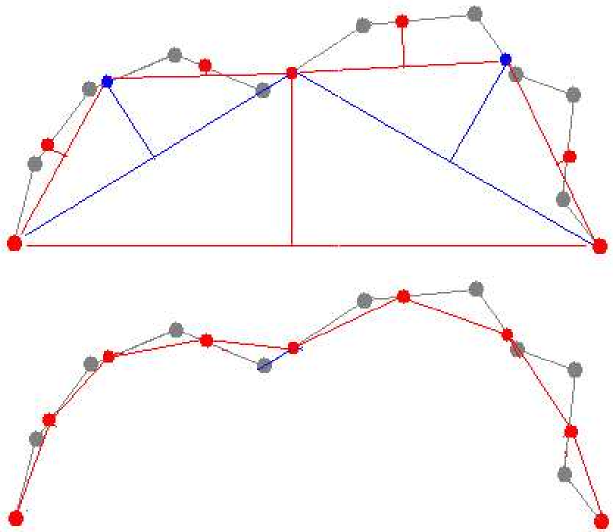
- multiscale methods (CAGD subdivision) using piecewise polynomials
- interpolatory
- shape preserving: monotonicity, convexity, etc
- optimal encoder for many classes (Besov balls):

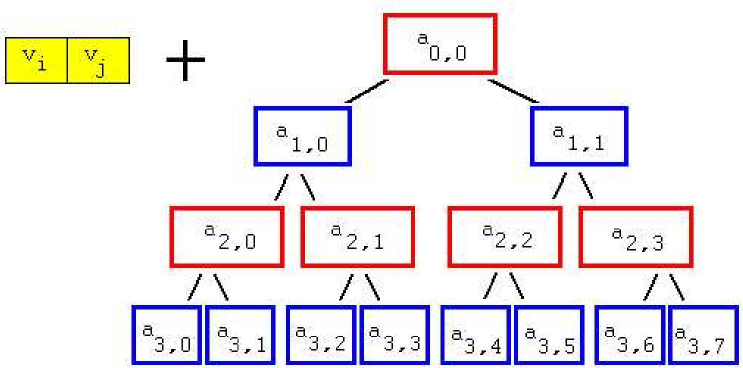
$$\delta_n(K_\alpha) \approx n^{-\alpha}, \quad K_\alpha := U(B^\alpha(L_\tau))$$

- K_1 includes all curves with finite arc length
- K_2 includes all convex curves

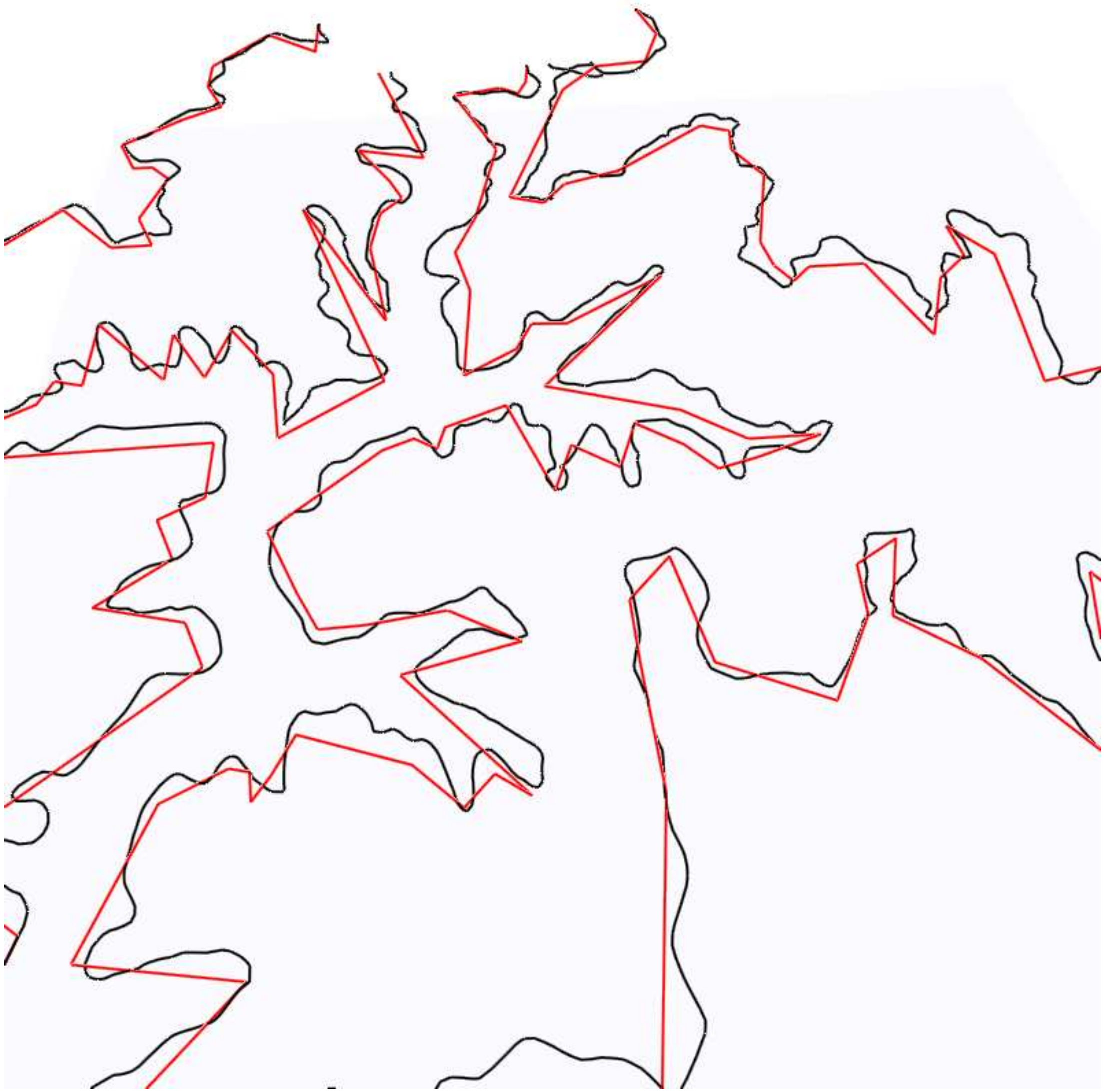
Simplest case - piecewise linear







Example: Level Curve Approximation



Further Efforts

- classify surfaces

Further Efforts

- classify surfaces
- determine Kolmogorov Entropy of these classes

Further Efforts

- classify surfaces
- determine Kolmogorov Entropy of these classes
- encode geometry

Further Efforts

- classify surfaces
- determine Kolmogorov Entropy of these classes
- encode geometry
- Burn In

Further Efforts

- classify surfaces
- determine Kolmogorov Entropy of these classes
- encode geometry
- Burn In
- Other Metrics

Predicting surface from wire mesh

- inpainting (nonlinear evolution equations)

Predicting surface from wire mesh

- inpainting (nonlinear evolution equations)
- constrained minimization

Predicting surface from wire mesh

- inpainting (nonlinear evolution equations)
- constrained minimization
- interpolation

Predicting surface from wire mesh

- inpainting (nonlinear evolution equations)
- constrained minimization
- interpolation
- constrained Delaunay

References

- A. Solé, V. Caselles, G. Sapiro, F. Arándiga, "Morse Description and Geometric Encoding of Digital Elevation maps," preprint.
- H. Carr, J. Snoeink, U. Axen, "Computing contour trees in all dimensions, Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms, 918-926 (San Francisco, January 9 -11, 2000).
- A. López, F. Lumbreras, J. Serrat, J. Villanueva, "Evaluation of Methods for Ridge and Valley Detection," IEEE Trans. Pattern Analysis and Machine Intelligence, **21**.4 (1999), 327–335.
- M. McAllister and J. Snoeyink, "Extracting consistent watersheds form digital river and elevation data," 1999 ASPRS Annual Conference.
- H. Edelsbrunner, J. Harer, A. Zomorodian, "Hierarchical Morse complexes for piecewise linear 2-manifolds," *SCG'01*, June 3-5, 2001, Medford, MA.

References (cont.)

- A. Cohen, W. Dahmen, I. Daubechies, and R. DeVore, Tree approximation and optimal encoding, ACHA **11**(2001), 192–226.
- R. DeVore, B. Jawerth, and B. Lucier Image compression through wavelet transform coding, IEEE Transaction on Information Theory **38**(1992), 719–746.
- R. DeVore, B. Jawerth, and V. Popov, Compression of wavelet decompositions, American Journal of Mathematics **114**(1992), 737–785.
- R. DeVore, L.S. Johnson, C. Pan, and R. Sharpley, Optimal entropy encoders for mining multiply resolved data, in "Data Mining II", N. Ebecken and C.A. Brebbia (eds.), WIT Press, Boston, 2000, pp. 73–82.
- M. Meyer, M. Desbrun, P. Schröder, A.H. Barr, “Discrete differential geometry operators for triangulated 2-manifolds, preprint.
- B. Sendov, *Hausdorff Approximations*, Kluwer Academic Publishers Group, Dordrecht, 1990, pp. 364.